

# Services et Web Services, Description de Service

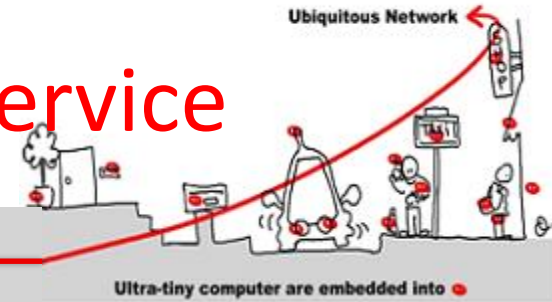
Jean-Yves Tigli

<http://www.tigli.fr>

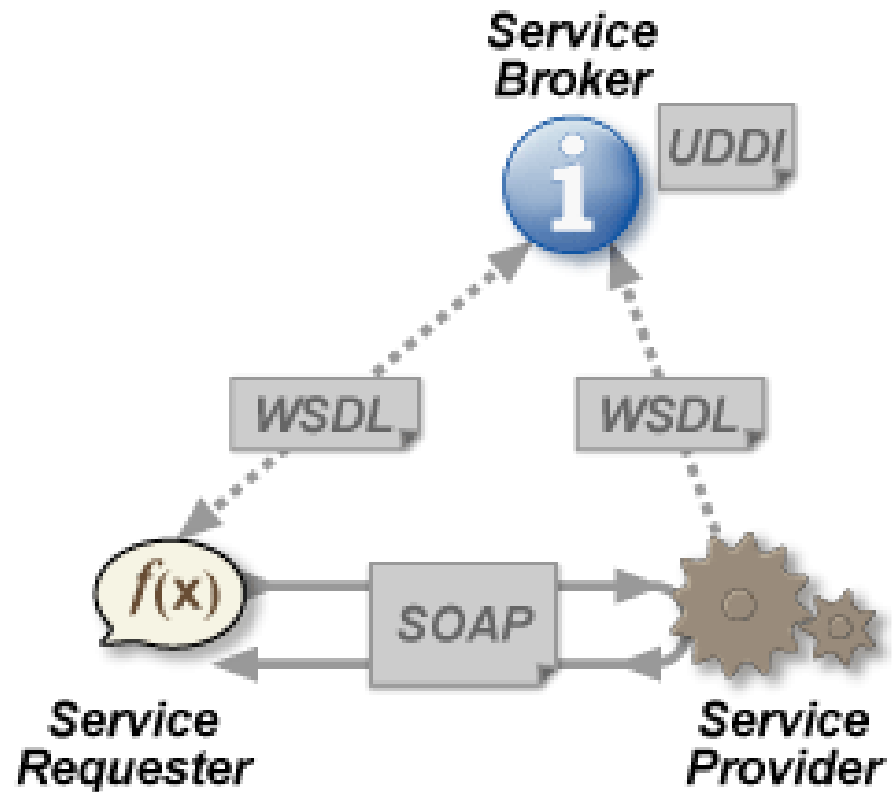
Polytech of Nice - Sophia Antipolis University

[Email : tigli@polytech.unice.fr](mailto:tigli@polytech.unice.fr)

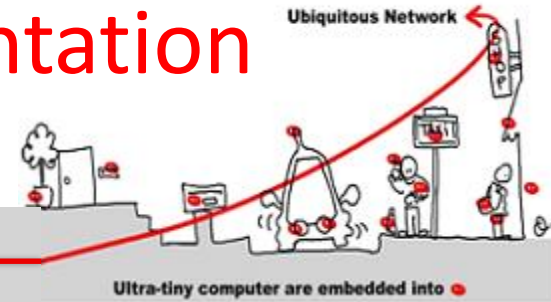
# Une représentation explicite du Service



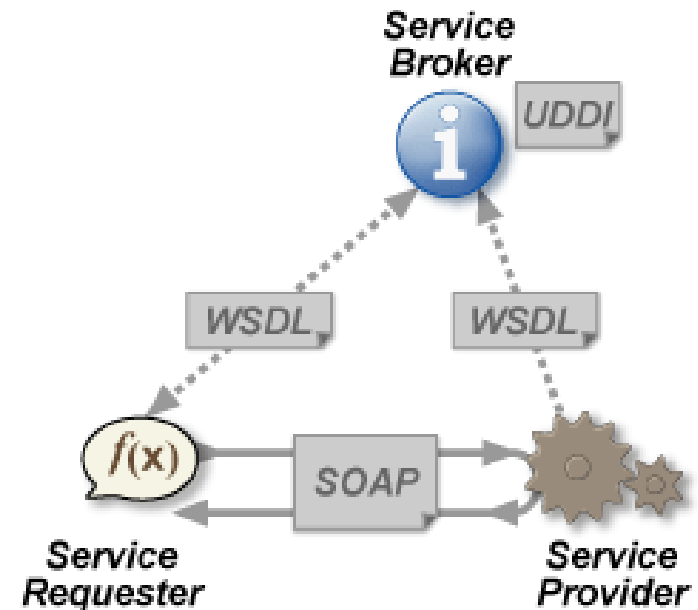
- Publication
- Découverte



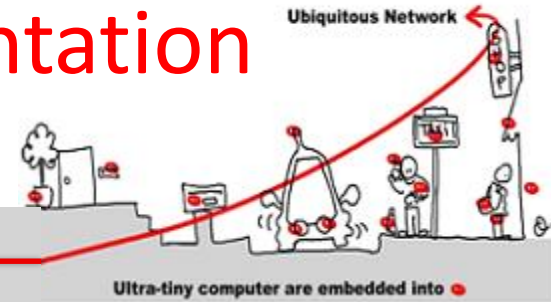
# Cycle de Vie WSOA : une représentation explicite du Service



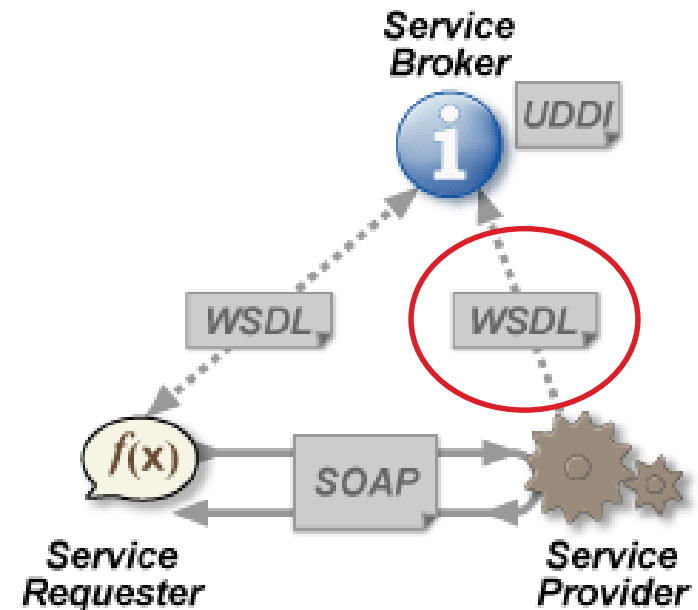
- Etape 1 : **Déploiement** du service Web
  - Dépendant de la plate-forme
- Etape 2 : **Enregistrement** du service Web
  - **WSDL** : description du service
  - Cf. **WS-\*** [www.w3c.org](http://www.w3c.org)
- Etape 3 : **Découverte** du service Web
  - Référentiels : DISCO (local), UDDI (global)
- Etape 4 : **Invocation** du service Web par le client
  - **WS-SOAP** (Cf. **WS-\*** [www.w3c.org](http://www.w3c.org))



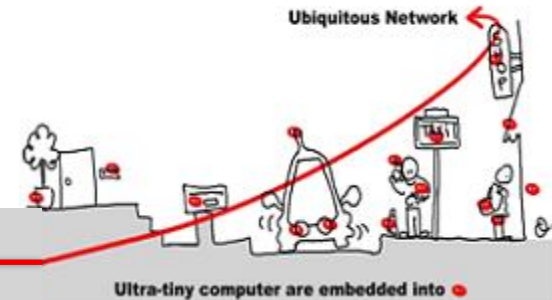
# Cycle de Vie WSOA : une représentation explicite du Service



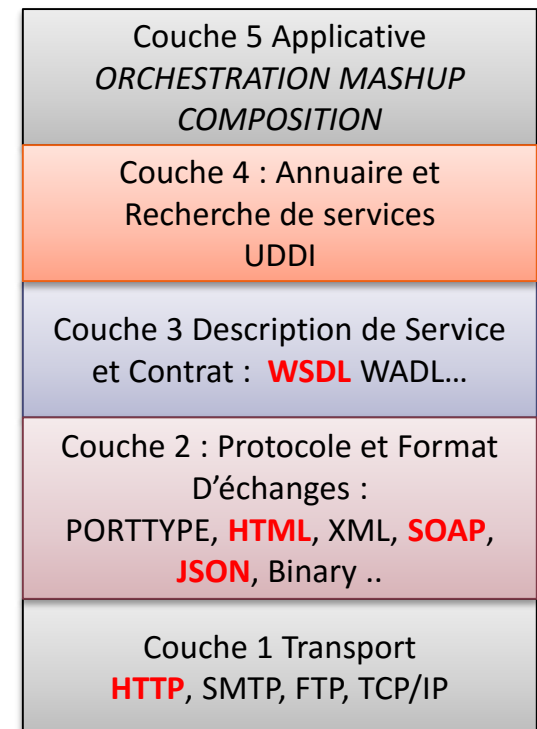
- Etape 1 : **Déploiement** du service Web
  - Dépendant de la plate-forme
- Etape 2 : **Enregistrement** du service Web
  - **WSDL** : description du service
  - Cf. WS-\* [www.w3c.org](http://www.w3c.org)
- Etape 3 : **Découverte** du service Web
  - Référentiels : DISCO (local), UDDI (global)
- Etape 4 : **Invocation** du service Web par le client
  - **WS-SOAP** (Cf. WS-\* [www.w3c.org](http://www.w3c.org))

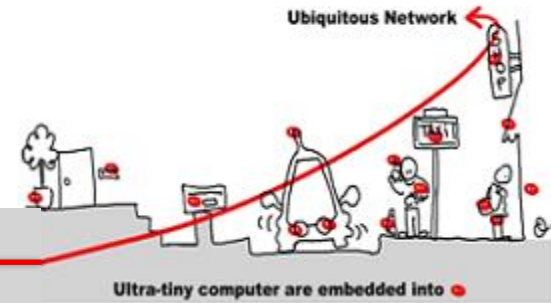


# Pour mémoire : Pile Protocolaire des Services Logiciels issus du Web



- Les principaux composants ou couches d'une pile de protocoles de services Web incluent :
- **Couche Transport**— assure la transmission des messages entre les applications
- **Couche Protocole et Format D'échanges** — encode et gère la séquence des messages échangés entre le service et son consommateur
- **Couche Description de Service et Contrat** — décrit le service fourni
- **Couche Annuaire et Recherche de Services**— centralise les services au moyen d'un registre commun





- WSDL (Web Service Description Language)

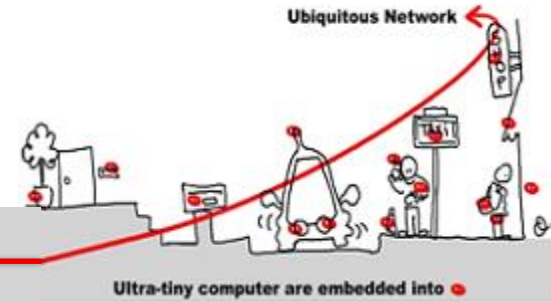
```

- <binding name="TestSoapBinding" type="tns:TestSoapPortType">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
- <operation name="Multiply">
  <soap:operation style="rpc" soapAction="http://soapinterop.org/Multiply" />
- <input>
  <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" use="encoded"
    namespace="http://soapinterop.org" />
  </input>
- <output>
  <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" use="encoded"
    namespace="http://soapinterop.org" />
  </output>
</operation>
- <operation name="Add">
  <soap:operation style="document" soapAction="http://soapinterop.org/Add" />
- <input>
  <soap:body use="literal" />
  </input>
- <output>
  <soap:body use="literal" />
  </output>
</operation>
</binding>
- <service name="MSInterop1DocAndRPCService">
- <port name="TestSoap" binding="tns:TestSoapBinding">
  <soap:address location="http://localhost/services/msInteropDocAndRpc" />
</port>
</service>
</definitions>

```

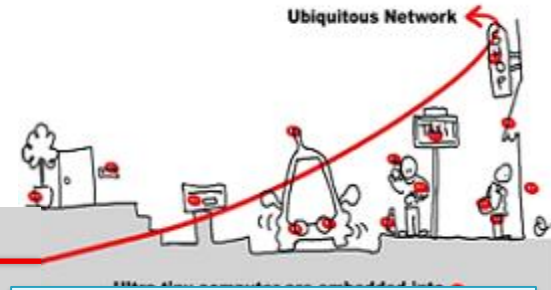
# DESCRIPTION DE SERVICE WS-SOAP

# WSDL



- Spécification
  - v1.1 pas « approuvée » par le W3C (note du 15-03-2001)
    - Soutenu par Ariba, IBM, Microsoft
  - v1.2 « Working Draft » du W3C (11-06-2003)
  - v2.0 recommandation du W3C (27-06-2007)
- Objectif
  - Interface publique d'accès à un Web Service
  - Comment communiquer pour utiliser le service (ensemble d'opérations et de messages abstraits reliés (bind) à des protocoles et des serveurs réseaux)
- Grammaire XML (schema XML)
  - Modulaire (import d'autres documents WSDL et XSD)
- Séparation entre la partie abstraite et concrète

# Les concepts de WSDL 1.1



- `<types>`
  - Contient les définitions des types (utilise un système de typage comme XSD)
- `<message>`
  - Décrit les noms et types d'un ensemble de champs à transmettre
    - Paramètres d'une invocation, valeur du retour, ...
- `<portType>`
  - Décrit un ensemble d'opérations et les messages impliqués (0 ou 1 en entrée, 0 ou n en sortie). Partie la plus importante
- `<binding>`
  - Spécifie une liaison d'un `<porttype>` à un protocole concret (SOAP1.1, HTTP1.1, MIME, ...). Un `portType` peut avoir plusieurs liaisons !
- `<port>`
  - Spécifie un point d'entrée (endpoint) comme la combinaison d'un `<binding>` et d'une adresse réseau
- `<service>`
  - Pour agréger un ensemble de ports

`<definitions>`

`<import>`

`<documentation>`

`<types>`

`<message>`

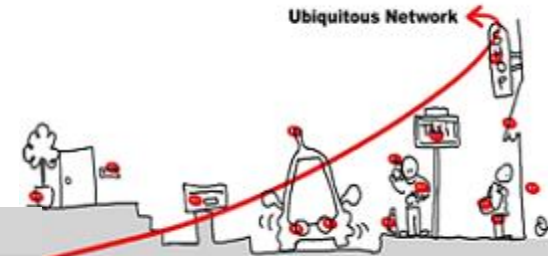
`<portType>`

`<binding>`

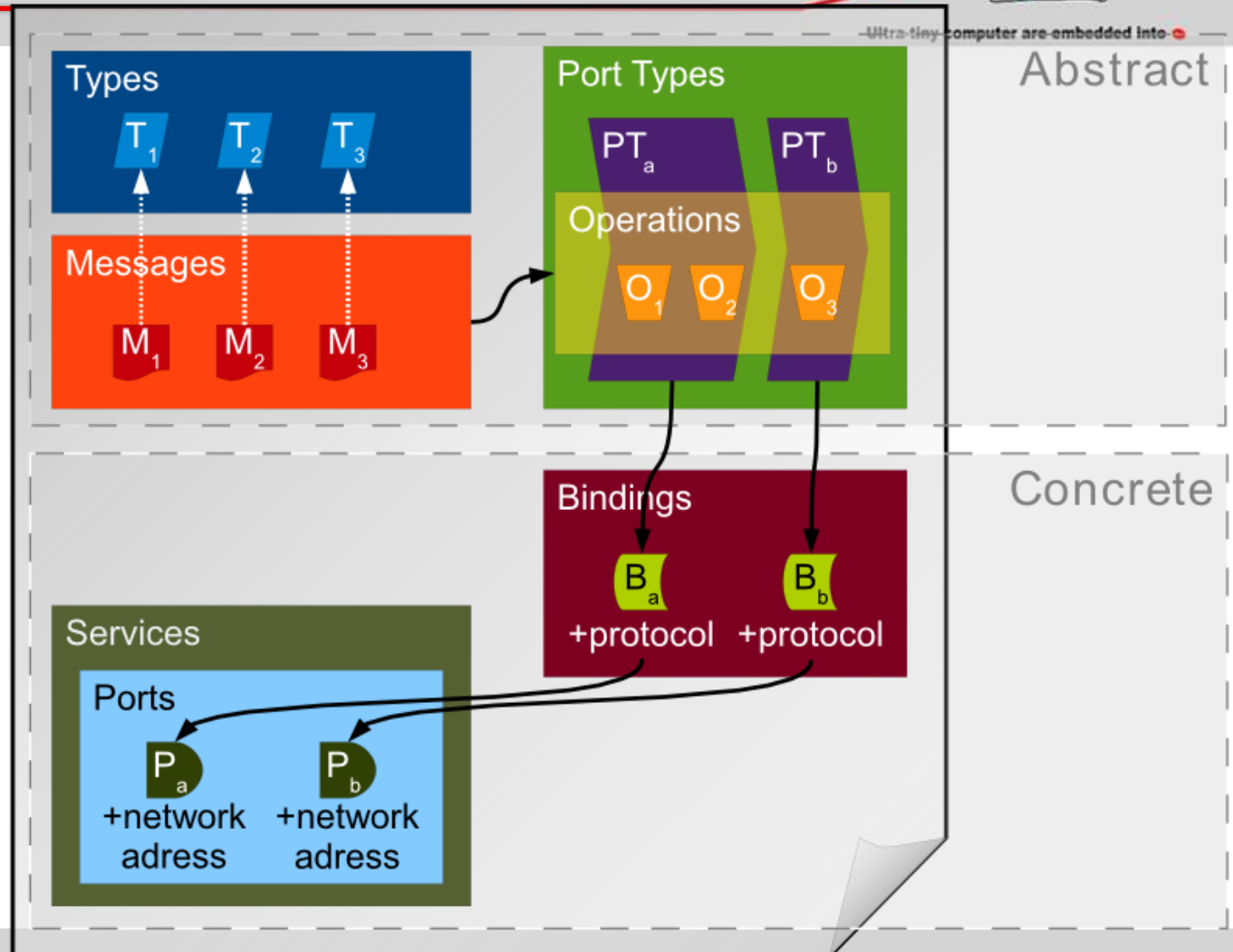
`<service>`



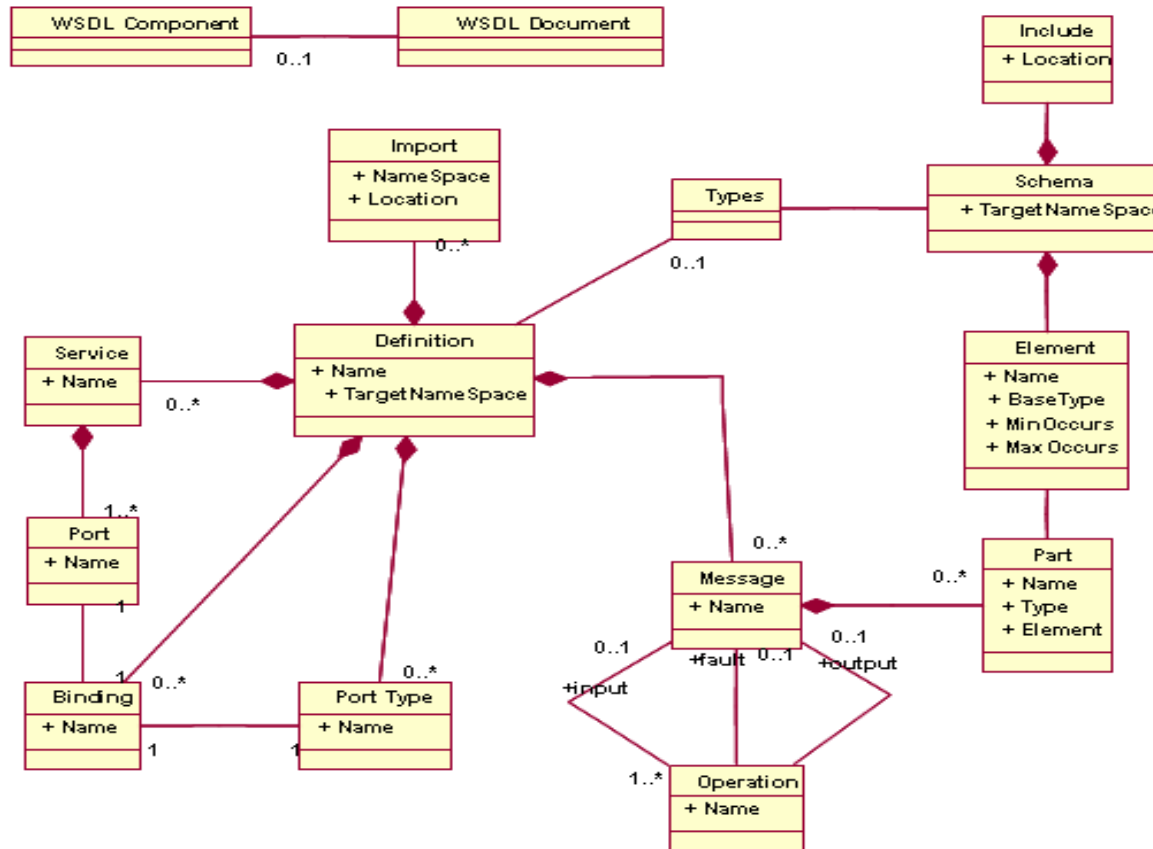
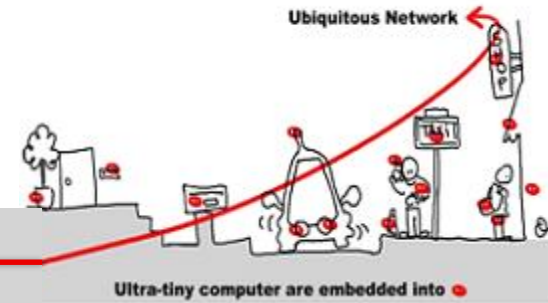
# Illustration de WSDL 1.1



- Proche du Modèle Générique ABC que nous verrons

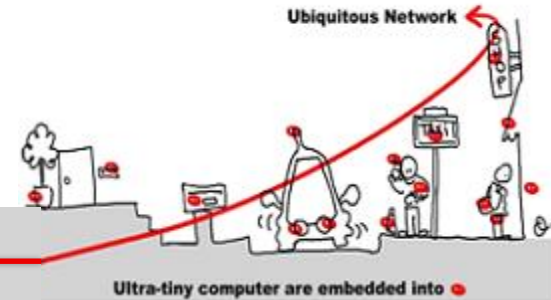


# Meta Modèle WSDL 1.1



Référence : <http://practices.modelbased.net/>

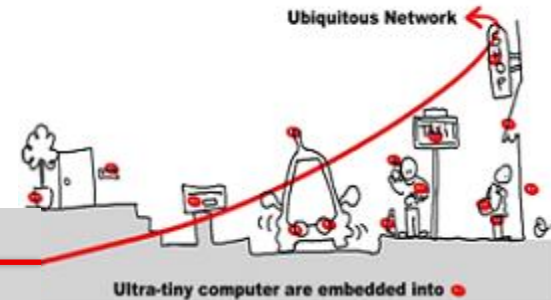
# Élément <types>



- Contient les définition de types utilisant un système de typage (comme XSD).
- Exemple

```
<!-- type defs -->
<types>
  <xsd:schema targetNamespace="urn:xml-soap-address-demo"
              xmlns:xsd="http://www.w3.org/1999/XMLSchema">
    <xsd:complexType name="phone">
      <xsd:element name="areaCode" type="xsd:int"/>
      <xsd:element name="exchange" type="xsd:string"/>
      <xsd:element name="number" type="xsd:string"/>
    </xsd:complexType>
    <xsd:complexType name="address">
      <xsd:element name="streetNum" type="xsd:int"/>
      <xsd:element name="streetName" type="xsd:string"/>
      <xsd:element name="city" type="xsd:string"/>
      <xsd:element name="state" type="xsd:string"/>
      <xsd:element name="zip" type="xsd:int"/>
      <xsd:element name="phoneNumber" type="typens:phone"/>
    </xsd:complexType>
  </xsd:schema>
</types>
```

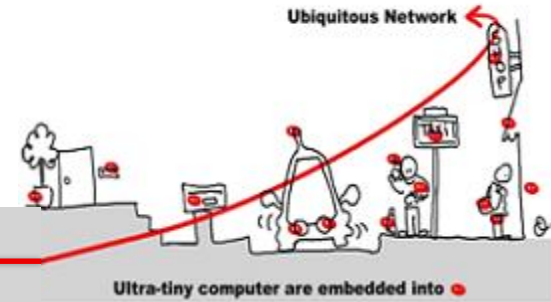
# Élément <message>



- Décrit les noms et types d'un ensemble de champs à transmettre
  - Paramètres d'une invocation, valeur du retour, ...
- Exemple

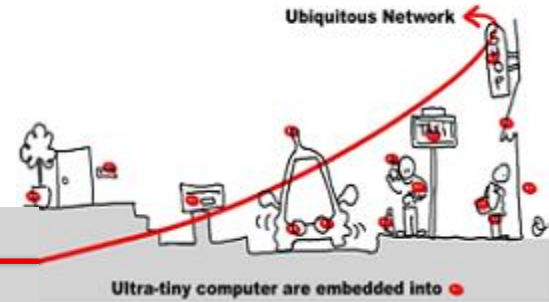
```
<!-- message declns -->  
<message name="AddEntryRequest">  
  <part name="name" type="xsd:string"/>  
  <part name="address" type="typens:address"/>  
</message>  
  
<message name="GetAddressFromNameRequest">  
  <part name="name" type="xsd:string"/>  
</message>  
  
<message name="GetAddressFromNameResponse">  
  <part name="address" type="typens:address"/>  
</message>
```

# Élément <porttype>



- Décrit un ensemble d'opérations (peut être vu comme une librairie, un module ou une classe)
- Plusieurs types d'opérations
  - One-way
    - Le point d'entrée reçoit un message (<input>) mais sans réponse
  - Request-response
    - Le point d'entrée reçoit un message (<input>) et retourne un message corrélé (<output>) ou un ou plusieurs messages de faute (<fault>).
  - Solicit-response
    - Le point d'entrée envoie un message (<output>) et reçoit un message corrélé (<input>) ou un ou plusieurs messages de faute (<fault>).
      - Binding HTTP : 2 requêtes HTTP par exemple
  - Notification
    - Le point d'entrée envoie un message de notification (<output>)
- Paramètres
  - Les champs des messages constituent les paramètres (in,out, inout) des opérations

# Exemple <porttype>



- Exemple

```
<!-- port type declarations -->
```

```
<portType name="AddressBook">
```

```
<!-- One way operation -->
```

```
<operation name="addEntry">
```

```
<input message="AddEntryRequest"/>
```

```
</operation>
```

```
<!-- Request-Response operation -->
```

```
<operation name="getAddressFromName">
```

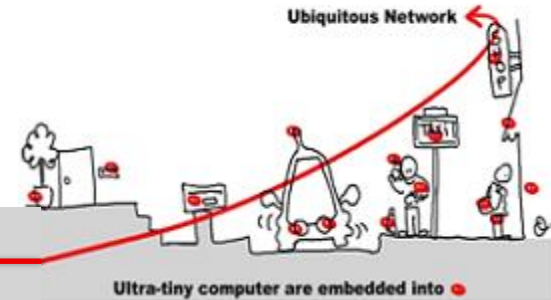
```
<input message="GetAddressFromNameRequest"/>
```

```
<output message="GetAddressFromNameResponse"/>
```

```
</operation>
```

```
</portType>
```

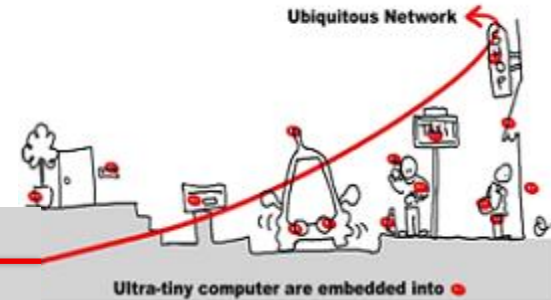
# Élément <binding>



- Spécifie une liaison d'un <portType> à un protocole concret (SOAP1.1, HTTP1.1, MIME, ...)

```
<binding name="AddressBookSOAPBinding" type="AddressBook">
  <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="addEntry">
    <soap:operation soapAction=""/>
    <input> <soap:body use="encoded" namespace="urn:AddressFetcher2"
      encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/> </input>
    <output> <soap:body use="encoded" namespace="urn:AddressFetcher2"
      encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/> </output>
  </operation>
  <operation name="getAddressFromName">
    <soap:operation soapAction=""/>
    <input> <soap:body use="encoded" namespace="urn:AddressFetcher2"
      encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/></input>
    <output> <soap:body use="encoded" namespace="urn:AddressFetcher2"
      encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/></output>
  </operation>
</binding>
```

# Élément <binding>

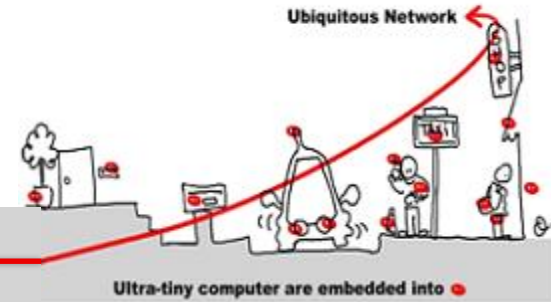


- Exemple SOAP / SMTP

```
...  
<binding name="StockQuoteSoap" type="tns:StockQuotePortType">  
<soap:binding style="document" transport="http://stockquote.com/smtp"/>  
<operation name="SubscribeToQuotes">  
<input message="tns:SubscribeToQuotes">  
<soap:body parts="body" use="literal"/>  
<soap:header message="tns:SubscribeToQuotes" part="subscribeheader"  
use="literal"/>  
</input>  
</operation>  
</binding>  
<service name="StockQuoteService">  
<port name="StockQuotePort" binding="tns:StockQuoteSoap">  
<soap:address location="mailto:subscribe@stockquote.com"/>  
</port>  
</service>  
</definitions>...
```



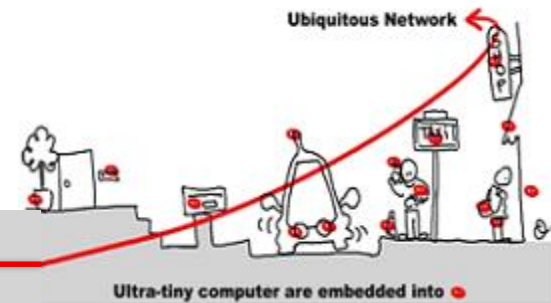
# Élément <service>



- Une collection de points d'entrée (endpoint) relatifs
- Exemple

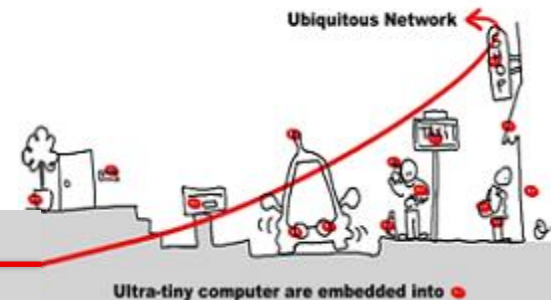
```
<?xml version="1.0" ?>
<definitions name="urn:AddressFetcher"
  targetNamespace="urn:AddressFetcher2"
  xmlns:typens="urn:xml-soap-address-demo"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns="http://schemas.xmlsoap.org/wsdl/">
  ...
  <!-- service decln -->
  <service name="AddressBookService">
    <port name="AddressBook" binding="AddressBookSOAPBinding">
      <soap:address location="http://www.mycomp.com/soap/servlet/rpcrouter"/>
    </port>
  </service>
</definitions>
```

# Il existe aussi un Langage de Description de Services REST



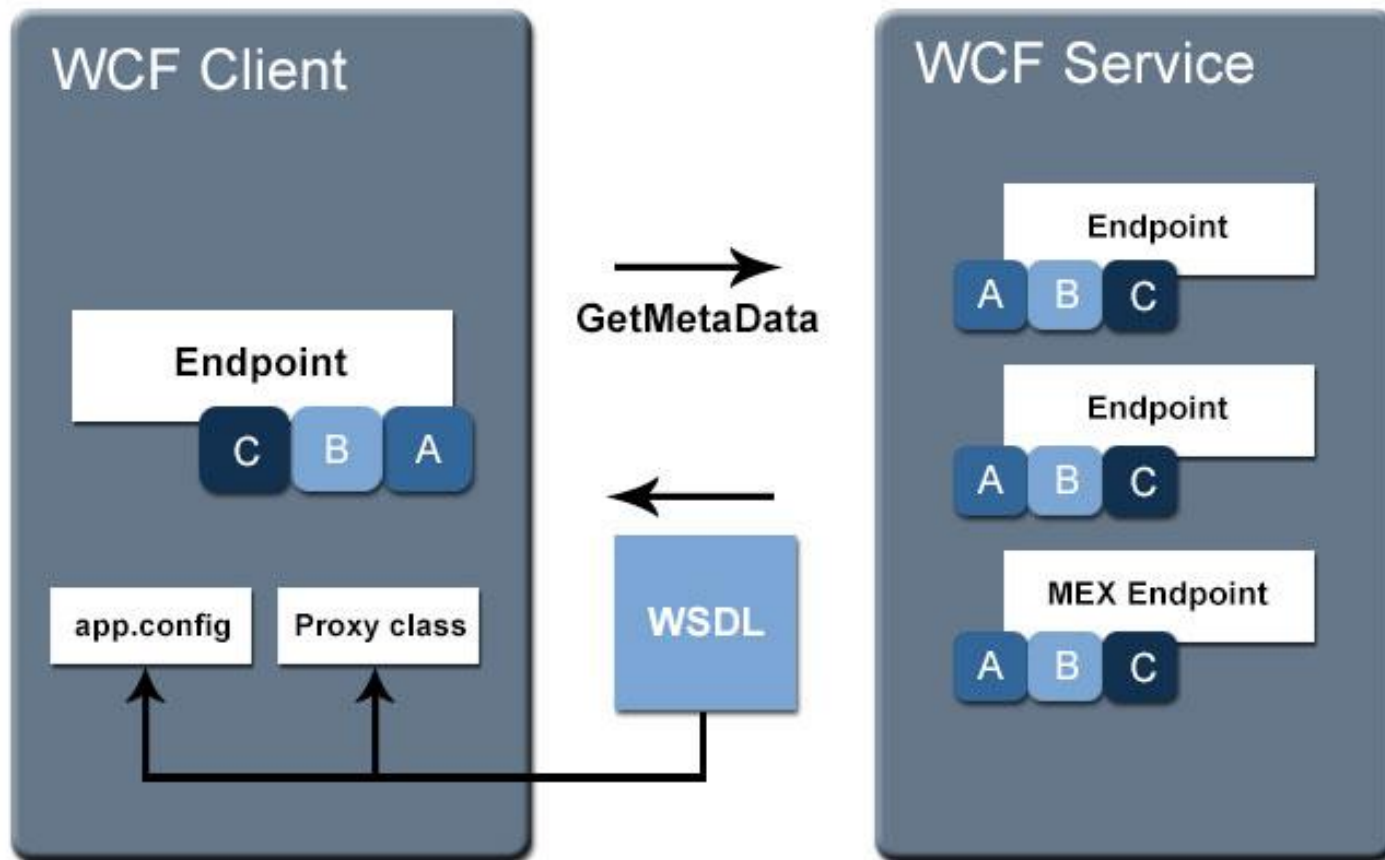
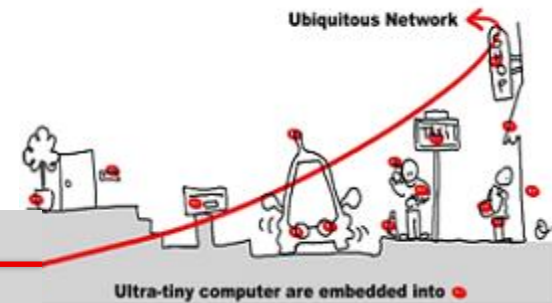
- WSDL (Web service description Language) : standard W3C (décrit maintenant du REST)
  - version 1.1
    - Pour décrire des web services SOAP. Basé sur XML.
    - (Microsoft and IBM)
    - <http://www.w3.org/TR/wsdl>
  - version 2.0
    - Pour décrire des web services SOAP et REST. Basé sur XML.
    - (Sun, Canon, IBM, WSO2)
    - <http://www.w3.org/TR/wsdl20/>

# Il existe aussi un Langage de Description de Services REST

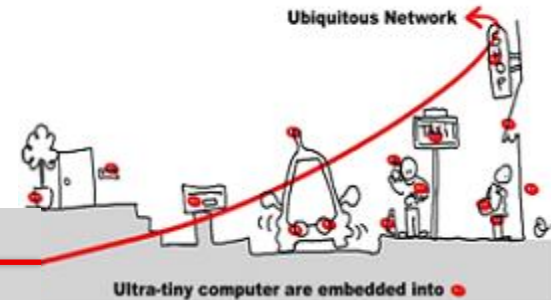


- WADL (Web Application Description Language) non encore standardise ...
  - Pour décrire des web services REST. Basé sur XML.
  - (Sun Microsystems)
  - <https://wadl.dev.java.net>
- D'autres initiatives générant un contrat et les métadonnées par introspection de code de librairie


# Introduction aux Metadata Exchange de WCF



# Les bindings pour MEX



```
<endpoint address="mex" binding="mexHttpBinding"  
contract="IMetadataExchange" />
```

- mexHttpBinding pour la publication HTTP. 
- mexHttpsBinding pour la publication HTTPS.
- mexNamedPipeBinding pour la publication de canal nommé.
- mexTcpBinding pour la publication TCP.